UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/411,792 | 10/01/1999 | David Alan Eward | 99-TK-238 | 8808 |

7590 08/25/2008

Lisa K. Jorgenson, Esquire
STMicroelectronics, Inc.
1310 Electronics Drive
Carrolton, TX 75006-5039

| EXAMINER | |
|---|---|
| VO, TED T | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2191 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 08/25/2008 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Application Number: 09/411,792
Filing Date: October 01, 1999
Appellant(s): EWARD ET AL.

Scott J. Gerwin
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 06/10/2008 appealing from the Office action mailed 01/10/2008.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

5,737,516 Circello et al. 4-1998

Hohl et al., "Debug Support on the ColdFire Architecture", Motorola Inc. (5-1996), pp. 1-10.

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-64 are rejected under 35 U.S.C. 102(b) as being anticipated by:

Hohl et el, "Debug Support on the ColdFire Architecture", Motorola (hereinafter: **MotorolaNPL**), and

Circello et al., US Pat. No. 5,737,516, assigee Motororla, Inc, submited by Appellants (hereinafter: **MotorolaPAT**).

Multiple Reference under MPEP 2131.01 is applied in this rejection:

> MPEP 2131.01 Multiple Reference 35 U.S.C. 102 Rejections:
> Normally, only one reference should be used in making a rejection under 35 U.S.C. 102.
> However, a 35 U.S.C. 102 rejection over multiple references has been held to be proper
> when the extra references are cited to:
> (A) Prove the primary reference contains an "enabled disclosure; "
> (B) Explain the meaning of a term used in the primary reference; or
> (C) Show that a characteristic not disclosed in the reference is inherent.

As per Claim 1: Motorola discloses,

"***At least one processor*** (MotorolaNPL: p. 2, Fig. 3: "Core");

***a debug circuit*** (MotorolaNPL: p. 2, Fig. 3: "Debug Module");

***a system bus coupling the processor and debug circuit*** (Conventional: all internal buses of a microprocessor. MotorolaNPL: p. 2, Fig. 3: connection buses); ***and***

***a communication link coupling the processor and debug circuit*** ((MotorolaNPL: p. 2, Fig. 3, or Fig. 4: such as K-Bus and PST),

***wherein the processor is configured to transmit to the debug through the communication link a plurality of bit values each representing a state*** (bit values of K-BUS and PST) ***of an operation*** (MotorolaNPL, See Fig. 4, p. 3, using PST: claim: '*processor is configured to transmit configured to transmit*); and further see p. 3, right column, paragraph started with, "To capture…")

***in the processor including at least an operand address that indicates a memory location at which an operand value is stored*** (MotorolaNPL: p. 6. Sample code of Fig. 7, shows "Inst address" such as 0001318: (claim: '*an operand address*') and "Instruction" such as mov.b d0, (-4,a6): (claim: '*operand value*'). Furthermore, see MotorolaPAT: Refer to FIG. 12, it shows generating data values for KADDR (i.e. Motorola: Inst Address, and claim: '*an operand address*') and KDATA (i.e. Motorola: Operand data, and claim: '*operand value*').

MotorolaNPL: see p.6, within the lines 4-5 of PST/DDATA/Description, operand values [3:0] and [7:4] **are displayed** on DDATA port. These operand values are stored at the address location 00001318.

MotorolaNPL: p. 3, right column, paragraph started with, "To capture…":

> To capture data and display it on the DDATA port, the WDDATA instruction fetches the **operand** defined in the effective address, then places the appropriate number of nibbles on the DDATA output pins, independent of any debug configuration.

According to the above portion, effective address: 00001318 presents in KADDR at the time. The address 00001318 indicates the memory location of fetched operand **d0** or **(-4,a6)**.

As per Claims 2-20: Because Appellants made claims 1-20 a group, claims 2-20 stand or fall together with the rejection of Claim 1.

As per Claim 21: Motorola discloses, "***A microcomputer implemented on a single integrated circuit*** (MotorolaNPL: ColdFire microprocessor) ***the microprocessor comprising*;**

***at least one processor*** (MotorolaNPL: p. 2, Fig. 3: "Core");

***a debug circuit*** (MotorolaNPL: p. 2, Fig. 3: "Debug Module");

***a system bus coupling the processor and debug circuit*** (Conventional: all internal buses of a microprocessor. MotorolaNPL: p. 2, Fig. 3: connection buses); ***and***

***a communication link coupling the processor and debug circuit*** ((MotorolaNPL: p. 2, Fig. 3, or Fig. 4: such as K-Bus. MotorolaPAT: see figure 2, all connections/bus between core 9 and the debug module 10),

***where the processor is configured to transmit to the debug through the communication link a plurality of bit values each representing a state of an operation in the processor*** (MotorolaNPL, See Fig. 4, p. 3, using PST: claim: '*processor is configured to transmit configured to transmit*); and further see p. 3, right column, paragraph started with, "To capture…") ***including at least one of:***

***an operand address that indicates a memory location at which an operand value is stored; and an operand value*** (MotorolaNPL: p. 6. Sample code of Fig. 7, shows "Inst address" such as 0001318: (claim: '*an operand address*') and "Instruction" such as mov.b d0, (-4,a6): (claim: '*operand value*'). Furthermore, see MotorolaPAT: Refer to Figure 12 that generate data values

for KADDR (i.e. Motorola: Inst Address, and claim: '*an operand address*') and KDATA (i.e. Motorola: Operand data, and claim: '*operand value*').

MotorolaNPL: see p6, within the lines 4-5 of PST/DDATA/Description, it **appears** that operand values [3:0] and [7:4] **are displayed** on DDATA port. These operand values are stored in the memory at the memory address location 00001318.

MotorolaNPL: p. 3, right column, paragraph started with, "To capture…":

> To capture data and display it on the DDATA port, the WDDATA instruction fetches the **operand** defined in the effective address, then places the appropriate number of nibbles on the DDATA output pins, independent of any debug configuration.

According to the portion, effective address: 00001318 presents in KADDR at the time, where 00001318 indicates a memory location of fetched operand **d0** or **(-4,a6)** which is an operand value.,

***where the processor is further to configured to transmit to the debug circuit: a program counter value indicating the program counter of the processor at a writeback stage of a pipeline of the processor a status indicating that a computer instruction is in the writeback stage is valid computer instruction*** (MotorolaPAT: See Figure 12, it shows a writeback stage of pipelines of the processor cores)***; a status indicating that the computer instruction in the writeback stage is a first instruction past an execute branch instruction*** (MotorolaNPL: see p. 6, values of PST)***; a status indicating a type of executed branch instruction and process identifier information of an executed instruction*** (MotorolaNPL: refer to p. 6: Fig. 7, PST/DDATA. In this particular case (coupled with Figure 12 of MotorolaPAT), the execution

fetches a branch at Inst Address = 00001326, and see the description of DDATA in PST/DDATA synchronization).

As per Claim 22: Regarding,

***"A Microcomputer comprising:***

***at least one processor***

***a debug circuit;***

***a system bus coupling the processor and debug circuit; and***

***means for transmitting to the debug circuit a plurality of bit values each representing a state of an operation in the processor including at least an operand address that indicates a memory location at which an operand value is stored***":

The claims have the claimed functionality corresponding to the functionality of Claim 1. Claims are rejected in the same reasons set forth in connecting to the rejection of Claim 1.

As per Claims 23-41: Because Appellants made claims 22-41 a group, claims 23-41 stand or fall together with the rejection of Claim 22.

As per Claim 42: Regarding,

***"A method for transferring information between a processor and a debug circuit over a communication link, the method comprising:***

***transmitting to the debug circuit a plurality of bit values each representing a state of an operation in the processor including at least an operand address that indicates a memory location at which an operand value is stored, and***

***transmitting a program counter value indicating the program counter of the processor***":

The claims have the claimed functionality corresponding to the functionality of Claim 1. Claims are rejected in the same reasons set forth in connecting to the rejection of Claim 1.

As per Claims 43-64: Because Appellants made claims 42-64 a group, claims 43-64 stand or fall together with the rejection of Claim 42.

**(10) Response to Argument**

Summary of the Motorola prior arts: Combining two Motorola's prior arts is permitted under MPEP 2131.01. The multiple reference is applied for supporting the Examiner's arguments during the pending period of the application.

Accordingly, the Motorola (MotorolaNPL) "Debug Support on the CoreFire Architecture" will be the primary prior art and be discussed in this Summary. Motorola (MotorolaPAT), "US Pat No. 5,737,516), would be referred to when characteristics not disclosed in MotorolaNPL is inherent.

Motorola (MotorolaNPL) discloses a debug module coupled with microprocessors and memory via an internal bus (See: p. 2: Fig. 3, and p.3 Fig. 4). It is known in the art that a microprocessor executes a sequence of instructions arranged in pipelines loaded from a memory. Conventionally, a data structure of memory comprises: Instruction Address and Instruction. It should be noted that "Instruction Address" is a program counter, where each value of this counter is an address of a set of bit data stored in the memory namely "Instruction".

Showing in p. 6, top left column of MotorolaNPL is sample code of a program sequence being carried out for debugging in the Debug Module of Fig. 4. The exampled program sequence as shown in p. 6 comprises "Inst Address", a program counter; and "Instruction", instruction information data:

| Inst Address | Instruction |
|---|---|
| 00001316 | movq #1, d0 |
| 00001318 | mov.b d0, (-4,a6) |
| 0000131c | pea (-68,a6) |
| 00001320 | lea Func2, a0 |
| 00001326 | jsr (a0) |
| 0000115c | mov.l d7, -(a7) |

According to the ColdFire debug module in Fig. 4 (i.e. Debug Module), debug information data which is stored in trace buffer, FIFO, is from K-BUS. That is the data which is generated from Pipelines (see MotorolaPAT, FIG. 12) and transmitted to Debug Module using KADDR and KDATA. Refer to MotorolaPAT (FIG. 12 and col. 6: lines 5-25), KADDR transmits "instruction address" by Instruction Address Generation, and KDATA transmits "operand data" by Operand Fetch.

Thus, if the sequence of the sample code above is brought into pipelines (MotorolaPAT), then the appearance of data in KADDR and KDATA should be "Inst Address" and "Instruction" (MotorolaNPL) respectively.

**(a):** MotorolaNPL: p. 3, right column, paragraph started with, "To capture…"

> To capture data and display it on the DDATA port, the WDDATA instruction fetches the **operand** defined in the effective address, and then places the appropriate number of nibbles on the DDATA output pins, independent of any debug configuration.

Motorola performs various debugging on a given program sequence. Fig. 4 in MotorolaNPL shows DDATA is controlled by "PST" and "CONTROL".

PST (execution status of the core) displays synchronization of DDATA, where each value of PST determines the number of bytes transferred to DDATA port.

CONTROL provides configuration status register (CSR). In MotorolaNPL, p.4, Tables 1-2, show types of data displayed on DDATA port based on the values in CSR (i.e. CONTROL).

In MotorolaNPL, p. 6, left column, it shows the synchronization of PST/DDATA, for outputting DDATA.

For example, within the synchronization of PST/DDATA, let the values of PST be **$8, 0,** then **0** (lines 3, 4, 5). Then, the data transmits on KDDAR and KDATA of Fig. 4 should include instruction addresses: 00001318 (corresponding to appellants' claims as: *transmit to the debug through the communication link a plurality of bit values... including at least an operand address*) and operand data in the instruction "mov.b d0, (-4,a6)", respectively (refer to MotorolaPAT, FIG. 12). With the correspondent values in PST = 0 (in line 4 and line 5), the displayed output DDATA is 4-bit [3:0] and 4-bit [7:4] of the instruction "mov.b d0, (-4,a6)". According to the byte-positions, these 4-bit bytes should be the operand data (-4,a6), and d0.

Thus, according to **(a)** above, address instruction "00001318" is the effective address, in which the WDDATA instruction fetches for the **operand data** with respect to 4-bit [3:0] and 4-bit [7:4]. The value "00001318" is the address of the operand data 4-bit [3:0] and 4-bit [7:4] in the instruction "mov.b d0, (-4,a6)".

Respectively, other instruction addresses and operand data of the sample code should be transmitted on KADDR and KDATA, according to PST/DDATA synchronization. It is customary for the skills in the art to discuss target instruction addresses at branches, JMP instructions, breakpoints, return addresses for their debug purposes. However, setting different values on CONTROL or PST means the ability of the debug module does more than certain discussions of prior arts.

A. **Argument of Claims 1-20**:

Appellants argued Motorola does not disclose transmitting to a debug circuit though a communication link a plurality of bit values each representing a state of an operation in the processor including "**at least on operand address that indicates a memory location at which an operand value is stored**". The Appellants alleged that nowhere does Motorola disclose "Operand Address". Appellants contended that KADDR might includes instruction addresses; however, that instruction address specifies the memory location of an instruction stored in memory, and they are claiming "Operand address" that specifies the memory location of an operand valued stored in memory.

Examiner's response: According to Appellants' specification (Brief: p3, within Claim 1, Appellants point to the specification at in page 7, lines 2-5, and lines 20-25), it generally discusses transferring to a debug circuit "debug information". Other generic discussions in the specification, at page 10, lines 6-10, commonly include "operand address information" and "operand data".

Based on the Appellants' description in the specification, the claims recite only about debug information without any further limitations for use of the "operand address" toward a practical manner. As provided with the examiner summary of the MotorolaNPL and MotorolaPAT above, Motorola shows the ColdFire architecture that is a single entity comprising a microprocessor core, memory, and debug module, internal buses. Motorola is focused on discussing branching because it is known that branching might causes errors. However, Motorola also discloses other things. For example, it shows a sample code and synchronized with PST (MotorolaNPL, p.6). This synchronization shows a data "00001318" presented in the KADDR at the time the microprocessor core fetches the instruction "mov.b d0, (-4,a6)".

It should be noted that the claimed language is about, "*transmit to the debug through the communication link a plurality of bit values… including at least an operand address.*". It is seen in the Motorola references that the address generator in the pipelines (shown in MotorolaPAT FIG. 12) generates an "**instruction address**" and places it in KADDR. The PST/DDATA synchronization shows that operand data of address"00001318" **is displayed as 4-bit [3:0] and 4-bit [7:4]**. Thus, the value of this **instruction address** is clearly "00001318". The operand data which is generated in the same time is clearly the data within "mov.b d0, (-4,a6)". Within the portion of MotorolaNPL, p. 3, mentioned in the summary of Motorola prior arts as **(a)** above, it has been proven that at the time the DDATA displaying [3:0] and 4-bit [7:4], the address "00001318" is in KADDR. Thus, the claimed recitation "*transmit to the debug*" reads on this value in the KADDR as the communication link to the debug module. Since 4-bit [3:0] and 4-bit [7:4] are the locations of the operand data which is stored in memory corresponding to the

address "00001318", the claim recitation "*at least an operand address*" reads on the address "00001318". Therefore, any argument for that MotorolaNPL or MotorolaPAT does not disclose OPERAND ADDRESS is inappropriate.

B. **Argument of Claim 21**:

Appellants argued Motorola does not disclose "**a microcomputer implemented on a single circuit**".

Examiner's response: Examiner disagrees: In this claimed argument, Appellants direct to a subject matter in the claimed preamble. It should be noted that the preamble is only intended to a scope of a claim. It should be noted that Claim 21 has its functionality corresponding to the functionality of claim 1. The preamble merely includes this subject matter. It should be noted that making a single or separate circuit, particularly for the preamble, does not cause the claim to be functionally different to any which does the same. Integrated circuit is well known. Motorola discloses "ColdFire processor architecture" (MotorolaNPL: P.2, start with section "2. ARCHITECTURE") which functionally acts as a single circuit. Because the preamble of the claim 21, merely addresses, "a microcomputer implemented on a single circuit", it reads on the ColdFire family of embedded processors (see p. 1), where Motorola discussed for a development that include a debug module within this architecture (See the section: 2. ARCHITECTURE, start from p. 2).

C. **Argument of Claims 22-41**:

Appellants group claims 22-41 separately. The argument for claims 22-41 is the same as the argument for Claim 1: They argued Motorola does not disclose **means** for transmitting to the debug circuit. The argument is that Motorola does not disclose "**including at least an operand address that indicates a memory location at which an operand value is stored**".

Examiner's response: The claims 22-41 recite "means" that appears corresponding to the "communication link" as of claim 1. With this limitation as well as appellants' argument, it does not produce any thing that is different from the argument in Claim 1. The patentability of the claims 22-41 must stand or fall together with the rejection of Claim 1.

D. **Argument of Claims 42-64**:

Appellants group claims 42-64 separately. The argument is that Motorola does not disclose "**including at least an operand address that indicates a memory location at which an operand value is stored**". The argument for claims 42-64 is the same as the argument for claim 1.

Examiner's response: The claims 42-64 recite the "method" that appears corresponding to the "communication link" as of claim 1. With this limitation and appellants' argument do not produce any thing that is different from Claim 1. The patentability of the claims 42-641 must stand or fall together with the rejection of Claim 1.

E. **Appellants argument for using multiple references in rejection of Claims 1-64 under 25 USC 102(b) is improper**.

Examiner's response: The use of multiple references is proper under MPEP 2131.01. In this case, MotorolaNPL and MotorolaPAT clearly discuss the same debug module. These two prior arts of Motorola are used for:

(A) Prove the primary reference contains an "enabled disclosure; "
(B) Explain the meaning of a term used in the primary reference; or
(C) Show that a characteristic not disclosed in the reference is inherent.

For example: MotorolaNPL shows the CoreFire core and the debug model as a single architecture. MotorolaPAT shows the pipelines that generate instruction address for KADDR and operand data for KDATA. It would be unnecessary for obviousness under 35 USC 103(a) in this case.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

TTV
August 11, 2008

/Ted T. Vo/

Primary Examiner, Art Unit 2191

Conferees:

/Wei Zhen/
Supervisory Patent Examiner, Art Unit 2191

/Eddie C. Lee/
Supervisory Patent Examiner, TC 2100